최종 보고서

헬스장 관리 프로그램

ME조(2011105108 홍진경, 2011105059 우병준, 2012105076 이현지)

소프트웨어 공학

2015 June 17

목차

[과제 소개 및 목표 1](#_Toc422340031)

[프로그램 소개 1](#_Toc422340032)

[과제 목표 1](#_Toc422340033)

[수행 일정 1](#_Toc422340034)

[팀원 소개 및 역할 분담 2](#_Toc422340035)

[요구사항 분석 및 SRS작성 3](#_Toc422340036)

[기존 요구사항 및 추가된 요구사항 3](#_Toc422340037)

[기존 SRS분석 및 개선된 SRS 분석 3](#_Toc422340038)

[test case 작성 6](#_Toc422340039)

[시스템 분석 8](#_Toc422340040)

[context-Diagram 8](#_Toc422340041)

[architcture design 분석 9](#_Toc422340042)

[class diagram 9](#_Toc422340043)

[품질향상도 분석 16](#_Toc422340044)

[Availability 16](#_Toc422340045)

[maintainability 17](#_Toc422340046)

[Safety 18](#_Toc422340047)

[secure 18](#_Toc422340048)

[Review 19](#_Toc422340049)

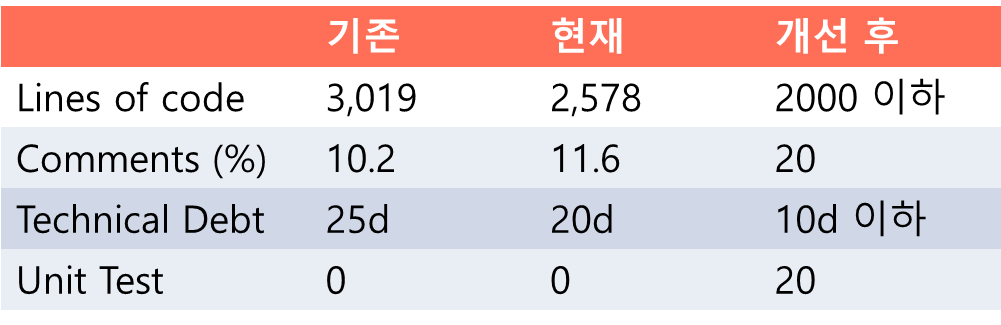
# 과제 소개 및 목표

## 프로그램 소개

* 이름: 헬스장 관리 프로그램
* 개요: 헬스장 관리자가 필요한 회원 관리, 운동프로그램, 운동기구 관리, 트레이너 관리를 할 수 있는 프로그램이다. 헬스장 회원도 같은 프로그램을 이용하여 자신의 운동 정보에 대해 파악할 수 있다.

## 과제 목표

1. 개선된 요구사항을 바탕으로 시스템에 필요한 기능을 추가하고 기존의 기능을 개선, 삭제한다.
2. 시스템의 분석 및 Refactoring과 Re-engineering을 통해 프로그램의 품질을 향상시킨다
3. 향후 프로그램 관리를 위해 유지보수성을 높인다.
4. 다양한 Testing을 통해 프로그램의 품질향상도를 확인한다.
5. SonarQube분석 결과를 토대로 다음과 같은 목표를 세웠다.



## 수행 일정

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | | Start  Week | End  Week | Week | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 요구 사항 |  | 1 | 2 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 요구사항 정리 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 추가 요구사항 분석 | 1 | 2 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| MRD 작성 | 2 | 2 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 분석 |  | 2 | 3 |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |
| SRS 작성 | 2 | 3 |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |
| 설계 |  | 3 | 5 |  |  | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |
| 시스템 설계 | 3 | 4 |  |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |
| DB 설계 | 3 | 4 |  |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |
| 화면 설계 | 3 | 5 |  |  | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |
| 인터페이스 | 3 | 5 |  |  | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |
| 개발 |  | 3 | 11 |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 환경 세팅 | 3 | 3 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |
| Test Case | 3 | 7 |  |  | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |
| Controller | 3 | 11 |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 관리자모드 | 3 | 7 |  |  | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |
| 운동기구관리 | 3 | 5 |  |  | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |
| 회원관리 | 5 | 7 |  |  |  |  | 1 | 1 | 1 |  |  |  |  |  |  |  |
| 운동프로그램관리 | 3 | 5 |  |  | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |
| 트레이너관리 | 3 | 7 |  |  | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |
| 회원모드 | 8 | 10 |  |  |  |  |  |  |  | 1 | 1 | 1 |  |  |  |  |
| 로그인 | 9 | 11 |  |  |  |  |  |  |  |  | 1 | 1 | 1 |  |  |  |
| UI개발 | 8 | 11 |  |  |  |  |  |  |  | 1 | 1 | 1 | 1 |  |  |  |
| TEST |  | 3 | 14 |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 모듈테스트 | 3 | 11 |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 통합테스트 | 10 | 11 |  |  |  |  |  |  |  |  |  | 1 | 1 |  |  |  |
| 최종평가 | 12 | 14 |  |  |  |  |  |  |  |  |  |  |  | 1 | 1 | 1 |

* 전반적으로 Incremental Development방식으로 개발을 진행하였다. 각자 맡은 기능에 대해 그 기능을 다 개발하면 다음 기능을 개발하며 진행.

팀원 소개 및 역할 분담

* 홍진경: 조장, 프로젝트 총괄 및 관리, 운동 프로그램, Database 개발
* 우병준: 운동기구 관리, 회원관리 기능 개발
* 이현지: 트레이너 관리, 회원모드기능 개발, 회의록 작성

# 요구사항 분석 및 SRS작성

## 기존 요구사항 및 추가된 요구사항

1. 기존 요구사항

헬스장 관리 프로그램을 도입하여 헬스장 관리인과 트레이너, 고객들에게 편의를 제공하고 싶다. 헬스장 관리인은 헬스장에 있는 회원, 운동기구, 운동프로그램, 트레이너 관리를 가능하게 하고 트레이너는 회원들의 운동상태를 관리하고 싶다. 고객들은 자신의 운동 플랜과 담당 헬스 트레이너 정보를 확인한다. 트레이너와 회원들은 회원가입을 통해 시스템에 등록할 수 있다.

1. 추가된 요구사항

관리자는 새로운 트레이너와 회원을 직접 추가하고 관리하고 싶다. 트레이너에게 회원들에게 맞는 운동기구와 프로그램을 배정할 수 있게 한다. 회원정보, 트레이너정보, 운동기구정보, 운동프로그램의 정보를 자세하게 볼 수 있었으면 좋겠다. 또, 손쉬운 관리를 위해 검색기능이 추가되었으면 좋겠다. 또, 불필요한 정보는 수집하지 않기를 원한다.

## 기존 SRS분석 및 개선된 SRS 분석

* **기존 SRS** \* ( ) 안의 숫자는 최대로 입력할 수 있는 칸의 수

1. Java를 이용하여 프로그램을 작성하고 txt파일 입출력을 통해 정보를 관리한다.
2. 로그인
   1. 로그인은 회원의 id로 한다.
   2. 관리자 및 트레이너는 옆의 관리자 버튼을 체크해야지만 관리자 패널에 접속할 수 있다.
3. 회원관리
   1. 회원정보: 회원번호[string(20)], 회원이름[string(20)], 전화번호[string(11)], 주소[char(50)], 등록 목적[string(50)], 운동가능 시간[string(50)], 배정트레이너[string(20)], 플랜[string(50)]
   2. 회원이 직접 회원가입을 해야 한다.
   3. 회원 가입과 동시에 트레이너 큐에 있는 트레이너를 차례로 부여받는다.
   4. 회원은 헬스클럽 출입 시 카운터의 관리자에게 출석체크를 받을 수 있다.
   5. 회원의 모든 정보가 한번에 리스트로 나타날 수 있도록 구현한다.
   6. 관리자와 트레이너만 조회가 가능하다.
4. 운동기구관리
   1. 운동기구정보: 이름[string(20)], 구입날짜[date], 구입가격[int(10)], 수량[int(3)]
   2. 한 화면에서 모든 운동기구의 정보를 리스트로 보여준다.
   3. 트레이너와 관리자가 조회할 수 있다.
   4. 관리자만 등록 및 삭제를 할 수 있다.
5. 운동프로그램관리
   1. 운동프로그램정보: 이름[string(20)], 운동효과[string(20)], 운동방법[string(20)]
   2. 운동 프로그램의 종류는 프로그램 내에서 수정이 불가능하고, txt파일 업데이트를 통해서만 추가, 삭제 및 수정이 가능하다.
   3. 관리자만 조회가능하다.
6. 트레이너 관리
   1. 트레이너정보: 이름[string(20)], 휴대폰번호[string(20)], 입사일[date], 주소[string(50)], 월급[int(10)], 특이사항[string(100)[
   2. 트레이너는 관리자가 추가 및 조회가 가능하다.

* **수정 및 추가된 SRS**

1. txt파일로 정보를 저장, 관리하는 경우 보안에 문제가 생기기에 MySQL database를 사용하여 보안성을 높인다.
2. 로그인
   1. 아이디와 패스워드를 통해 로그인을 한다. (패스워드 추가)
   2. 로그인 시 로그인 정보(관리자, 트레이너, 회원)에 따라 그에 맞는 화면을 보여준다. (기존의 체크박스 삭제)
3. 회원관리
   1. 회원정보: 이름(10자), 휴대폰 번호(000-0000-0000), 키(Int), 몸무게(Int), 체지방량(Int), 근육량(Int), 등록날짜(yyyy-mm-dd), 등록기간(Int), 생년월일(yyyy-mm-dd), 주소(200자), 배정트레이너(10자), 배정프로그램(10자), 특이사항(200자), 등록여부(Boolean)
   2. 회원 등록은 회원이 작성한 정보를 바탕으로 관리자만 할 수 있다.
   3. 회원의 로그인용 아이디는 회원의 휴대폰번호(11~12자리)이고, 초기 비밀번호는 생년월일 8자리로 한다. 차후에 비밀번호 변경 가능.
   4. 관리자에 의해 담당 트레이너가 부여되고, 트레이너에 의해 운동프로그램이 배정된다.
   5. 관리자와 트레이너만 열람가능하다.
   6. 해당 칼럼을 선택 후 상세보기 버튼을 클릭하면 회원 정보를 상세하게 볼 수 있다. 또, 회원정보에 변화가 있을 시 상세보기 패널에서 수정이 가능하다.
   7. 등록날짜와 등록기간을 자동으로 계산하여 만료일을 알려준다.
   8. 회원 정보 리스트 아래에 있는 검색 창에 카테고리를 선택하고 검색할 수 있다.
4. 운동기구 관리
   1. 운동기구: 이름(10자), 구입날짜(yyyy-mm-dd), 수량(Int), 파손유무(boolean), 비고(200자)
   2. 운동기구는 관리자에 의해 추가, 수정 및 삭제가 가능하다.
   3. 관리자와 트레이너만 열람할 수 있다.
   4. 운동기구 상세보기 클릭 시 그 운동기구에 대한 상세 내용이 나오며 관리자는 수정 버튼을 통해 수정이 가능하고 파손유무를 체크박스를 통해 관리할 수 있다.
   5. 운동기구 정보 리스트 아래에 있는 검색 창에 카테고리를 선택하고 검색할 수 있다.
5. 운동프로그램 관리
   1. 운동프로그램: 이름(10자), 난이도(10자), 운동부위(10자), 특이사항(200자)
   2. 운동프로그램은 관리자와 트레이너에 의해 추가, 수정 및 삭제가 가능하다.
   3. 운동 프로그램 상세보기 버튼을 누르면 정보를 상세히 볼 수 있고, 수정이 가능하다.
   4. 운동프로그램 리스트 아래에 있는 검색 창에 카테고리를 선택하고 검색할 수 있다.
6. 트레이너 관리
   1. 트레이너: 이름(10자), 휴대폰번호(000-0000-0000), 입사일(yyyy-mm-dd), 주소(200자), 월급(Int), 특이사항(200자)
   2. 트레이너 등록은 트레이너가 작성한 정보를 바탕으로 관리자만 할 수 있다.
   3. 트레이너의 로그인용 아이디는 트레이너의 휴대폰번호(11~12자리)이고, 초기 비밀번호는 입사일로 한다. 차후에 비밀번호 변경 가능.
   4. 관리자만 조회 가능하다.

## test case 작성

1. 운동기구 관리

* 운동기구 추가

**Input:** 운동기구 명(‘줄넘기’), 수량(3), 가격(10000), 구입 날짜(2015-02-10), 비고(‘아주 튼튼’)

**Test procedure:** input을 이용하여 commodity 객체 생성, guiProcess.add(commodity), 다시 getCommodity를 하여 객체를 받아와 출력

**Expected output:** 줄넘기, 3, 10000, 2015-02-10, 아주 튼튼

* 운동기구 검색

**Input**: (운동기구명, 줄넘기)

**Test procedure:** 데이터베이스에서 검색한 결과를 백터객체로 돌려 받고 해당 정보를 출력한다. 위에서 추가된 줄넘기 객체를 사용.

**Expected output**: 줄넘기, 3, 10000, 2015-02-10, 아주 튼튼

* 운동기구 삭제

**Input:** 1, commodity

**Test procedure:** delete(1, ‘commodity’), seleteCommodity를 이용하여 해당 튜플 검색

**Expected output:** null

1. 운동프로그램 관리

* 운동프로그램 추가

**Input:** 운동프로그램의 이름(‘요가’), 운동 부위(‘전신’), 난이도(‘초급’), 비고( )

**Test procedure:** input을 이용하여 program 객체를 생성 후 FileManager의 add기능을 수행한다. 그 후 getProgram을 이용하여 해당 정보를 불러와 출력한다.

**Expected output:** 요가, 전신, 초급,

* 운동프로그램 수정

**Input:** id(1), 운동 명(‘필라테스’), 운동부위(‘전신’), 난이도(‘중급’), 비고(‘요가대체’)

**Test procedure:** 요가가 저장된 튜플의 id값을 받아오고, 위의 정보로 객체를 만든 후, updateProgram을 이용하여 수정한 후, getProgram을 시행하여 출력한다.

**Expected output:** 필라테스, 전신, 중급, 요가대체

* 운동프로그램 삭제

**Input:** 1, program

**Test procedure:** delete(1, program), seleteProgram()을 이용하여 해당 튜플검색.

**Expected Output:** null

1. 트레이너 관리

* 트레이너 등록

**Input:** 트레이너 id(1), 입사일(2015-01-12), 전화번호(010-1111-1111), 월급(150), 나이(28), 주소(대구), 로그인 아이디(123), 비밀번호(123)

**Test Procedure:** 위의 정보를 이용하여 trainer객체를 생성 후, add(trainer)를 수행, getTrainer()를 이용하여 해당 정보를 다시 객체로 받아온 후 출력

**Expected Output:** 1, 2015-01-12, 010-1111-1111, 150, 28, 대구, 123, 123

* 트레이너 삭제

**Input:** 1, trainer

**Test Procedure:** delete(1, ‘trainer’), seleteTrainer를 이용하여 해당 튜플 검색

**Expected output:** null

# 시스템 분석

## context-Diagram

헬스장 관리 시스템

보안 시스템

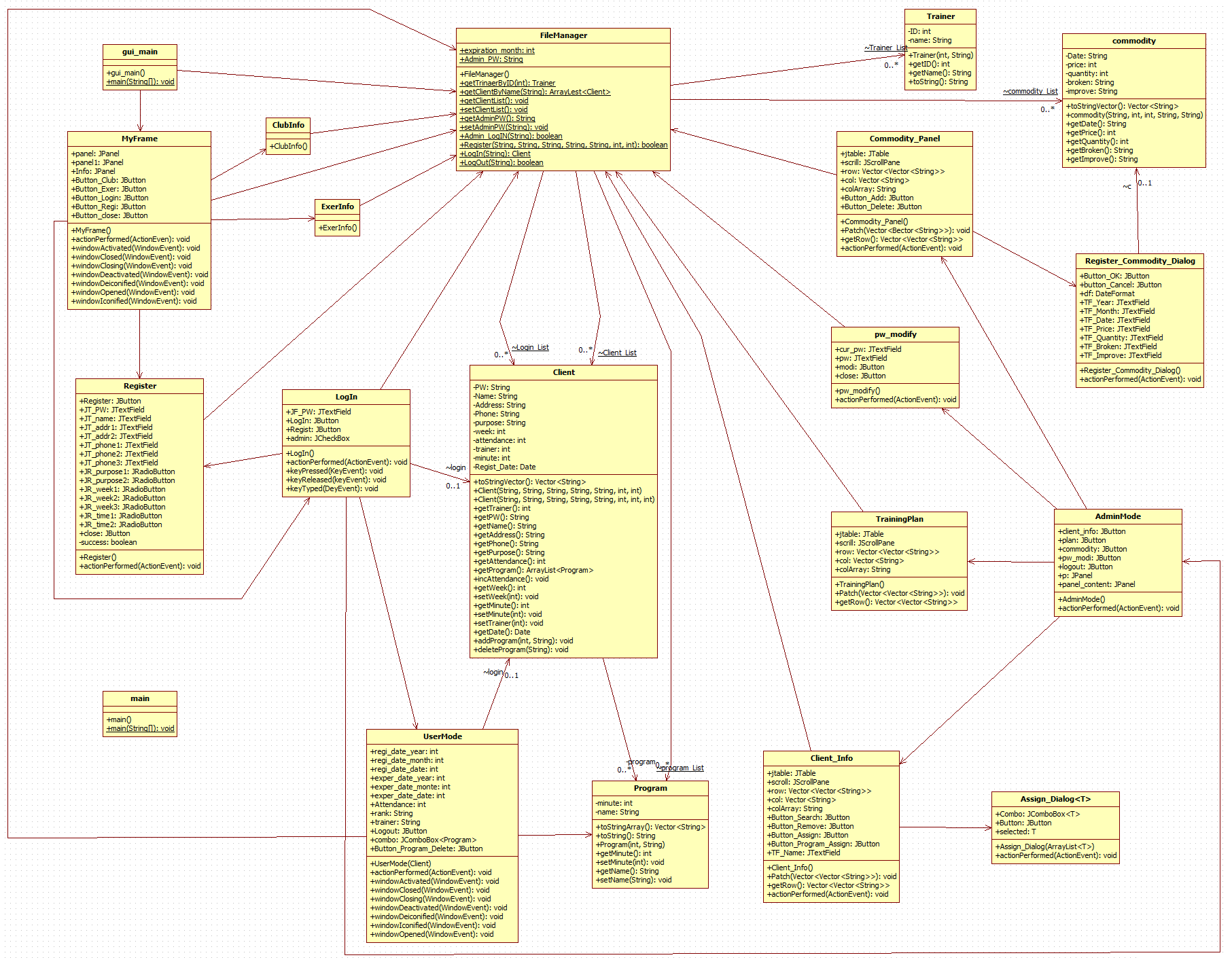
이 시스템은 외부 시스템과의 연관성이 적은 시스템이다. 따라서, 헬스장에 등록된 회원 및 트레이너의 정보의 보안을 유지해줄 시스템만 필요로 한다.

## architcture design 분석

  
현재 구현되어 있는 프로그램은 사용자가 로그인 하면, 요청된 패널(관리자용, 트레이너용, 회원용)을 볼 수 있고, 그에 따라 데이터를 열람, 추가, 삭제 등을 할 수 있다. 요청된 패널은 View에 해당되며, Trainer, Client, Commodity, Program Model에 있는 내용을 볼 수 있으며, 데이터의 직접적인 제어는 Controller에서 이루어진다. 사용자는 데이터 베이스에서 데이터가 어떻게 처리되는지 알 필요도 없고, 사용자 패널에서 데이터 베이스 처리기능을 분리해야 유지, 보수가 쉬울 것이다. 따라서 MVC아키텍처를 이용하여 사용자 인터페이스와 데이터 베이스 로직을 분리하여 설계할 것이다.

## class diagram

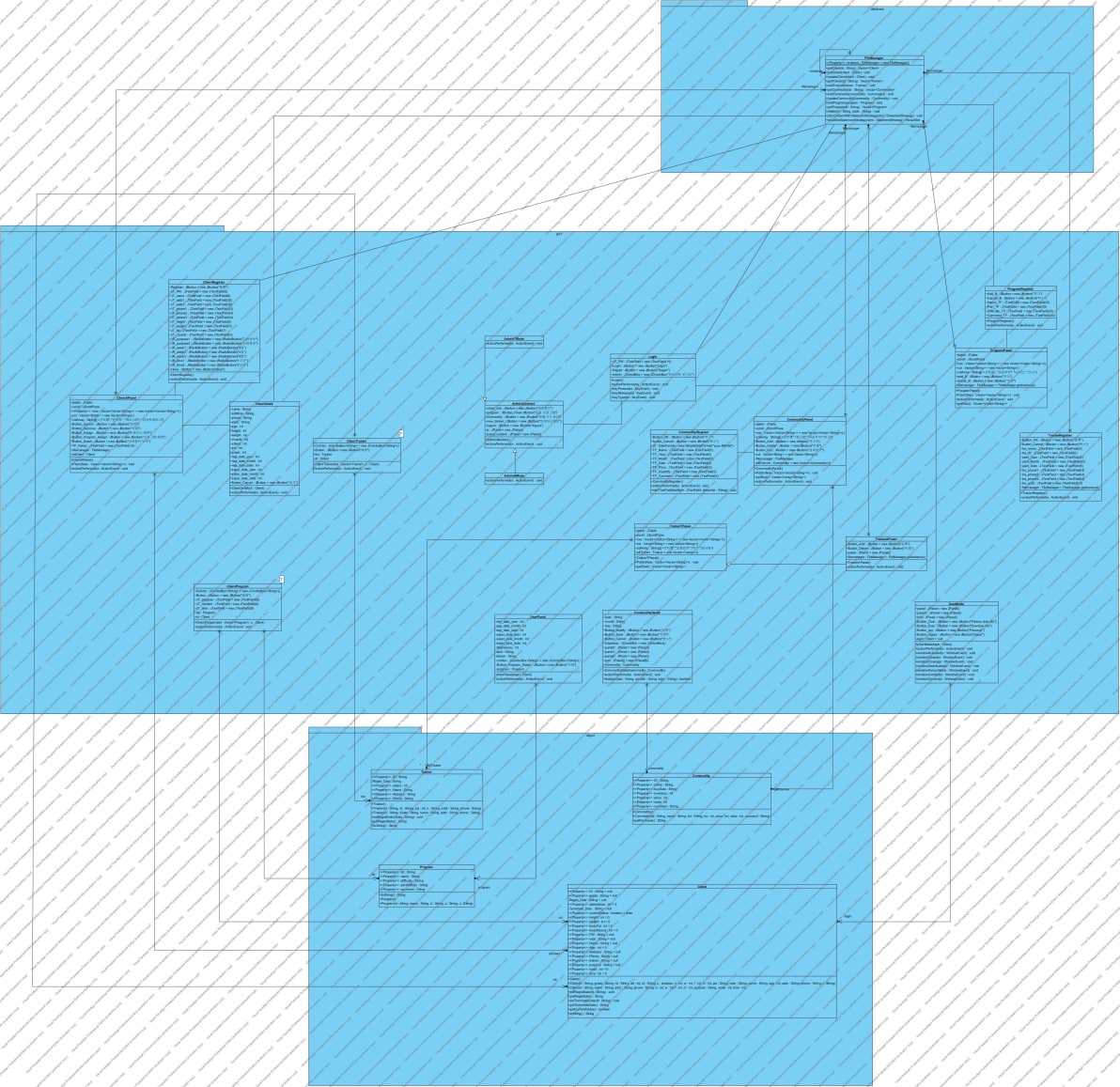
1. Class Diagram
2. 기존의 클래스다이어그램



* 문제점: 기능적으로 연관된 클래스들이 많으나 모두 따로 구현되어 있어서 중복되는 코드가 많고 어떠한 아키텍처 패턴에도 해당되지 않는다. 코드 내에서도 한 메소드 안에 여러 가지 기능이 구현됨에 따라 유지 보수성이 떨어진다.

1. 수정, 추가 후 클래스다이어그램

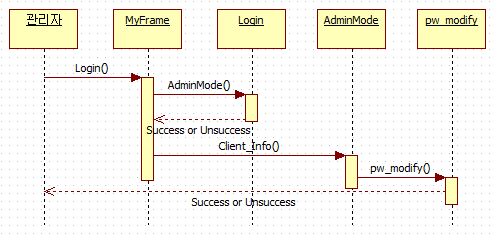
* 전체적으로 MVC 아키텍처 패턴을 사용함으로써, 다음과 같이 패키지를 구분하여 상위에서부터 Controller, View, Model의 기능을 수행하게 된다.
* GUI class는 GuiProcess의 메소드를 이용하여 FileManager의 데이터베이스 메소드에 접근할 수 있다.



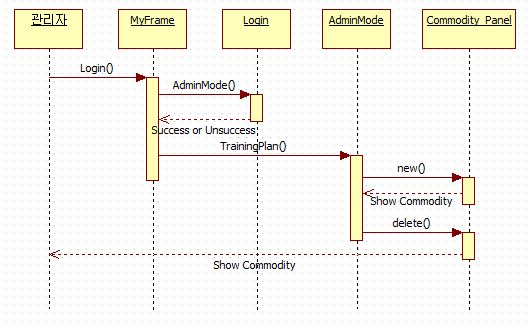
1. Sequence Diagram

* 기존의 시퀀스다이어그램

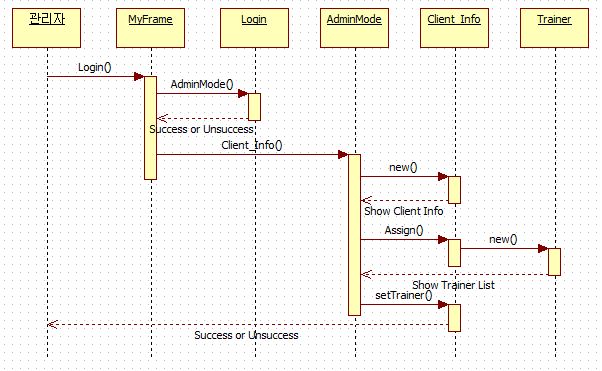
1. 관리자- 비밀번호변경



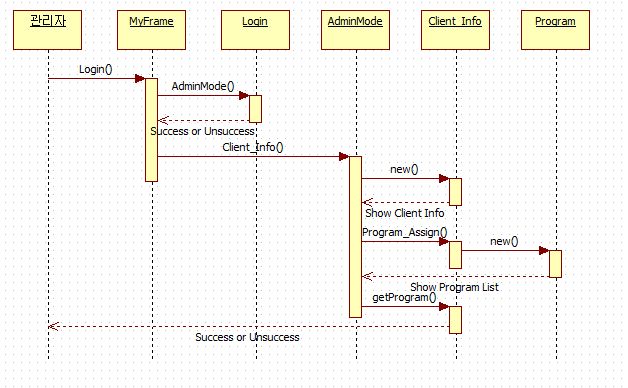
1. 관리자- 추가, 삭제



1. 관리자- 트레이너배정

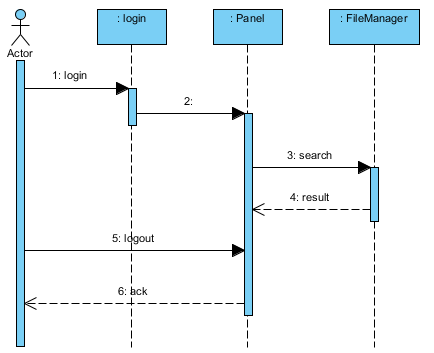


1. 관리자- 운동 프로그램 배정



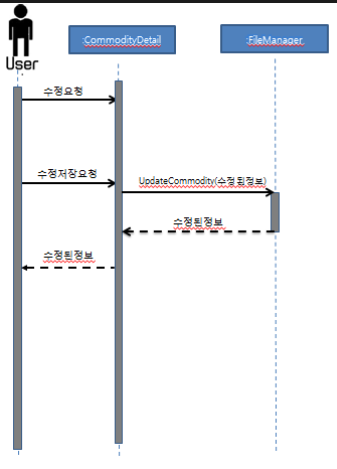
* 추가, 수정된 시퀀스다이어그램

1. 검색



회원, 운동프로그램, 운동기구, 트레이너를 검색할 수 있는 기능을 추가하였다.

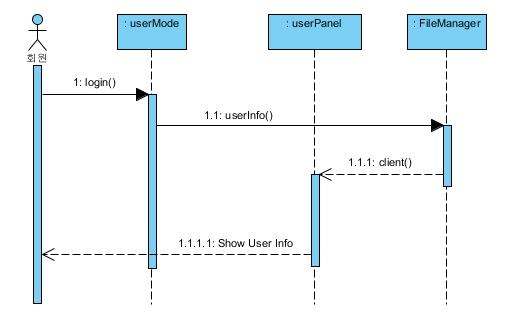
1. 정보수정



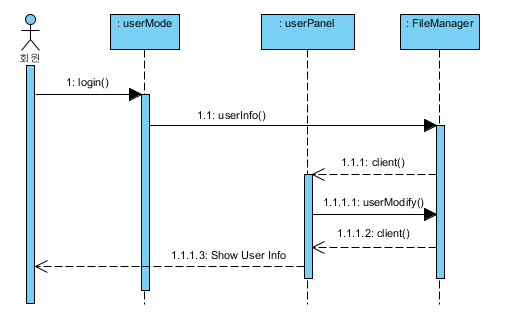
Admin이 운동기구정보, 트레이너정보, 회원정보를 수정 할 수 있다.

1. UserMode

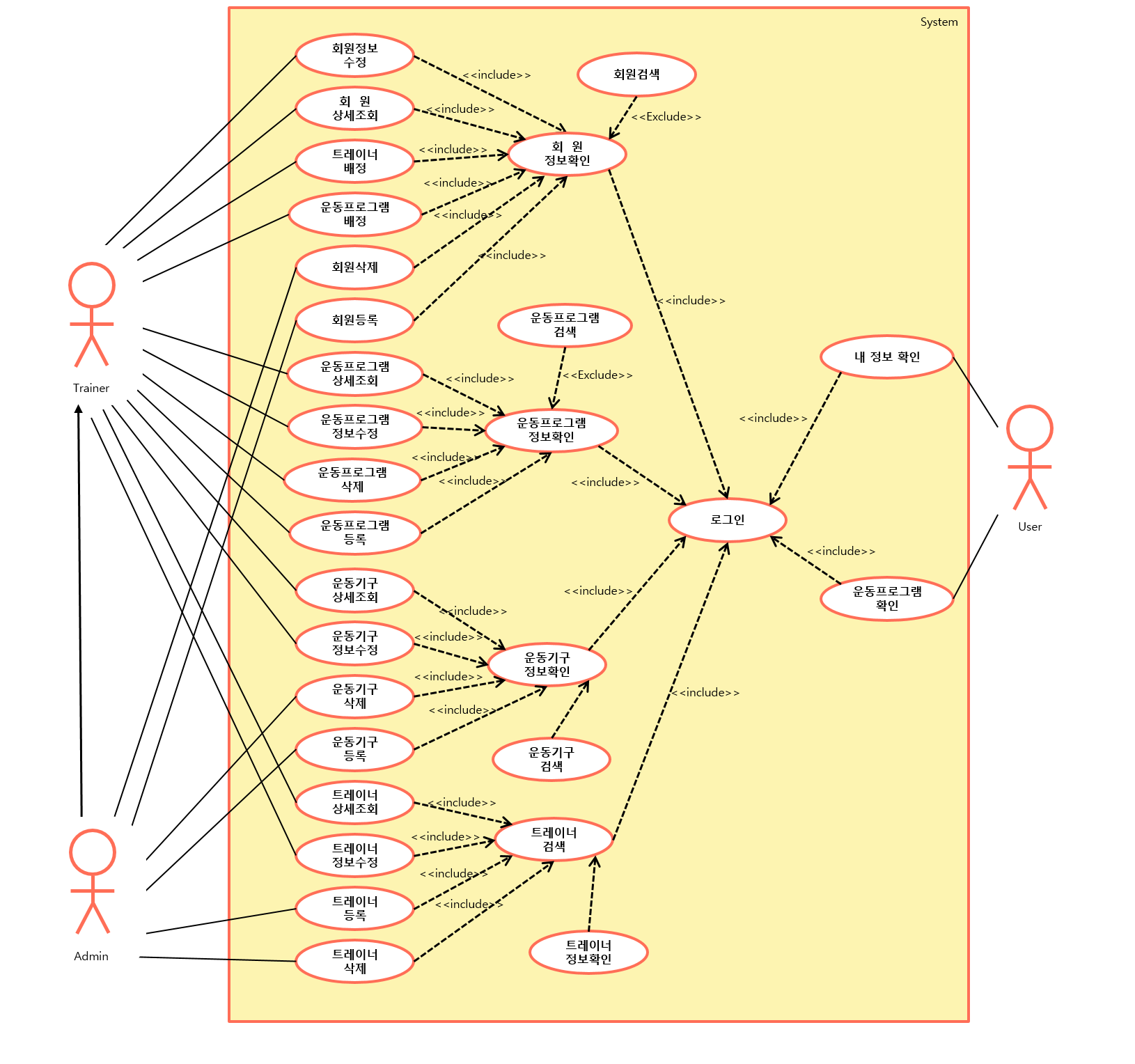
3-1. 회원정보보기: 회원이자신의정보(개인정보, 배정트레이너, 배정프로그램)를 확인하는 기능



3-2. 회원정보수정: 회원이 자신의 정보를 수정하는 기능



1. Use-Case Diagram



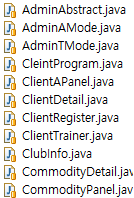
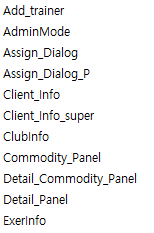
# 품질향상도 분석

## Availability

1. 38개의 클라이언트가 접속 시 현저히 속도가 저하되는 것을 발견하였다. 따라서 최대 동시 접속가능자수는 38명이다.

## maintainability

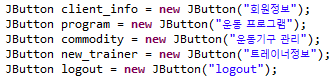
1. 기존 프로젝트의 클래스이름의 경우, 특별한 규칙이나 순서가 없어 프로그램 수정 시 어떤 클래스 파일인지 잘 구별되지 않았다. 그래서 [관련 있는 기능][용도]로 클래스 명을 수정하여 클래스의 Readability를 높였다.

1.

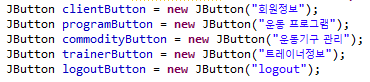
(변경 전, 변경 후)

1. 기존 프로젝트를 개발 시 여러 개발자가 각자 자신만의 언어로 변수를 선언하고 메소드를 선언하여 프로젝트 내에 변수명과 메소드명이 일관성이 없었고, 변수와 메소드 사용시 불편함이 많았다. 그래서 [관련 있는 기능][용도]로 통일성을 주어 프로젝트 코드를 이해하기 쉽게 하였다.

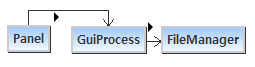
\* 변경 전



\* 변경 후



1. 기존의 프로젝트의 경우, 각 기능을 수행하는 GUI에서 직접적으로 Filemanager의 instance를 참조하여 기능을 수행하였다. 즉, GUI와 직접 Sql 연산을 수행하는 Filemanager와 연관성이 높아 유지, 보수 시에 문제가 있었다. 그래서 GUI와 Filemanager를 연결하는 GuiProcess 클래스를 새롭게 생성하여 GUI와 DB연산의 연관성을 낮추어 주었다.



1. 기존의 SQL Query를 수행하던 FileManager의 경우 복잡한 연산을 하는 method가 많았는데, 이 method들을 한 메소드가 하나의 기능을 수행할 수 있도록 Re-factoring하였다.
2. 구현된 기능에 대한 JUnit 테스트를 추가하여 코드 변경 시 정상적으로 작동함에 대해 자동화된 테스트가 가능하도록 하였다.
3. 이외에도 코드 Re-factoring을 하여 유지보수성을 높였다.

## Safety

1. 기존 시스템의 경우, 헬스장의 헬스장 내에 있는 운동기구의 파손 여부를 확인할 수 없는 시스템으로, 운동기구의 문제가 생겨도 체크할 수 없었다. 개선된 프로그램의 경우, 트레이너와 관리자가 관리할 수 있어 파손이 있는 즉시 관리자가 조치를 취하여 헬스장을 이용하는 회원들이 운동기구의 파손으로부터 당할 수 있는 피해를 줄일 수 있다.

## secure

1. 기존 시스템에는 로그인 시 아이디만 입력 받아 로그인을 하였다. 따라서 로그인 시 아이디만 안다면 누구나 접근이 가능하였고, 개인정보 탈취의 위험이 있었다. 그래서 아이디와 비밀번호를 함께 입력 받아 로그인을 하므로써, 보안성을 높였다. 또, 기존의 시스템의 경우, 최고 관리자도 마찬가지로 아이디를 입력받고 옆에 관리자 로그인에 체크를 하여 관리자 페이지에 접근할 수 있었다. 이 경우, 해커가 관리자의 아이디만 안다면 모든 정보를 탈취해 갈 수 있었으므로, 관리자 로그인 체크박스를 없애고 아이디, 비밀번호를 이용하여 로그인할 수 있게 하였다.
2. 기존 시스템은 txt파일의 암호화되지 않은 입출력으로 헬스장 내의 모든 정보를 관리하였다. 누구나 정보가 저장되는 곳의 위치만 알면 손쉽게 접근하여 정보를 얻을 수가 있어서 보안에 취약하였다. 그래서 외부 서버에 있는 데이터베이스를 이용하여 정보를 관리하고, 저장하도록 변경하였다.

# Review

1. 개선사항

* SonarQube분석결과



**→** 중복되는 라인의 수는 적어졌지만, 새로운 기능을 추가하고 클래스의 분리,

메소드의 분리 등으로 새로운 코드가 많이 추가되었다.

**→** Technical Debt은 10d까지 줄이지는 못했지만 근접하게 다가갔다.

* UI개선이 더 필요하다.
* Availability 측면: 다수의 동시접속을 가능케 하기 위해 서버가 다수의 클라이언트를 처리할 수 있도록 해야한다.
* Security 측면: txt파일 사용에서 DB사용으로 보안성을 강화하였지만, 서버의 데이터베이스를 보호할 수 있는 방법도 필요할 것이다. 또, 암호를 더욱 복잡하게 작성하도록 만들어 쉽게 비밀번호를 알 수 없도록 만든다.